4. ネットワーク形データベース

古幡 博（慈恵医大） 福本一朗（東大）
上野晴樹（青山学院大） 斉藤正男（東大）

4.1 ネットワーク形データベース

ネットワーク形DBSの代表例としてはCODASYLのDBTG提案があげられる。この提案はDB用共通言語を目指した提案であり、これを完全に実現しているシステムは未だ発表されていないようである。しかし、この提案を基礎にして、その部分的な達成に成功したシステムは数多く発表されている。IDSII, DMS 1100, また本邦におけるRAPID（富士通）, PDM（日立）などなどネットワーク型の例である。後述するようにこれらは親言語方式であり、DBTG提時に既にNIPS／TFS（DCA, IBM）, TDMS（Sys. Dev. Corp.）などの独立言語方式によるDBも存在していた。ここではDBTG提案を中心に述べ、必要に応じRAPIDの手法等を含め説明する。

4.2 歴史

ネットワーク形の最初のシステムはIDS（GE）（1963年）であるが、同時にDBSとしても最初のものである。DBSはアプリケーションプログラムからデータを独立させて、記憶容量の軽減、アプリケーションプログラム作成時の手順の省略化、検索ルーチンの簡素化、などを実現させようというものであるから、ネットワーク形独自の歴史はないが、主にファイル処理、検索という立場を強く意識した立場で開発されたものにネットワーク型のDBが多い。DBTG提案はIDSの影響を強く受けながら、DB用の共通言語システムを目指して、1966年から作業が始められ、1969, 1971年の二度の提案報告を経て、1975年にはCOBOL-DB機能へと結びついてきている。

4.3 使用計算機

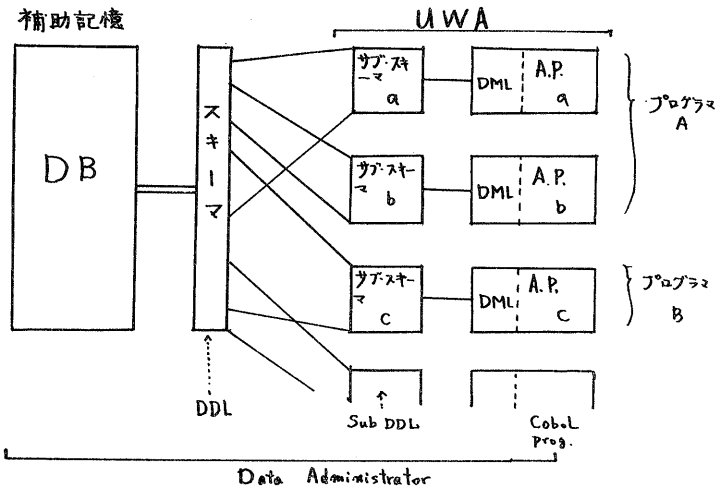
DBTG提案は計算機に何ら規定を加えていないが、実現されているシステムは中型機以上と見られる。CPU記憶容量が262kB以上を持つOSの上で動かされている。

4.4 特徴

ネットワーク型の代表であるDBTG提案は以下の特徴を持っている。
①ホスト言語方式、（データベースの取扱うデータにアクセスするには、COBOL, FORTRAN等の手続き型言語を必要とする方式）。②記述は叙述式言語形態をとる。③2つのDDLを使う。1つはDB全体のスキーマを記述するものであり、もう一つは、アプリケーションプログラムとDBとの間にあって、アプリケーションプログラムに必要なサブDBを示すために記述するものである。後者をサブ・スキーマDDLと言う。④DBSの変更機能が無い。⑤PLの機能がある。⑥データ管理者が必要。⑦どちらかというと利用者はプログラマである。⑧後述する如く集団関係（セット）を陽に定義する

4.5 全体の構成

DBTGが行っている提案によれば、DBとユーザプログラマとの関係は図－1のような形式になる。ユーザーの立場からDB内のデータを使用したプログラムを実行しようという状況を考えてみよう。まずプログラマは必要とするデータを



図－1　DBの全体構成

明らかにし、データ管理者と相談する。データ管理者（DA）はDB全体を熟知している者であり、①スキーマを作成し、②DBに対する利用状況を把握し、③その記録を作成し、④機密チェックを行い、⑤そして今問題にしているプログラマに対する領域の割当てを行う。従ってDAはプログラム、あるいはその業務を円滑かつ、機密保護条件を満足するように、行なわせるため、プログラマと相談しながら、サブ・スキーマを作成する。サブ・スキーマはプログラム固有のデータ構造をもち、スキーマ（全体を記述する）の一部を割合自由に写像したものである。しかも、この写像は単なるコピーでなく、構造、データ、PL Key などDAと相談の上、スキーマと違った形にすることができる。さて、プログラマはサブ・スキーマを手に入れても、なお、それを利用するにはDML（Data Manipulation Language）を自己のProg.中に含めなければならない。DMLはData baseに対する変更、更新、検索などの手順を示すものである。この操作言語とスキマ定義のDDLは実行時に強い関連を持ち、ある操作に対するLock機能はDDLでスキーマ又はサブ・スキーマ内に記述されている。あるいは集団関係を消去する場合にも、DDLで消去の仕方を指定することができる。DMLで、サブ・スキーマを通して始めて、プログラムはBase内のデータを手にすることができる仕組みになっている。この場合プログラムはCOBOLでもFORTRANでも良い。（ただしFORTRAN用のDMLはまだ発表されていない。）

4.6 集団関係とその実現

上述の如く、利用者はDB内にデータが物理的にどのように記憶されているか関知せず、論理的なデータ構造だけを念頭において処理Prog.を作成することができる。しかし、この論理的な構造にも種々の型があり、しばしば木構造とか、網構造とか呼ばれる。表題のネットワーク形と称したのはこの網構造のことである。一般にDB内の構造は表1の如く表わせる。この表中cが、データ構造を様々な形にしているものであり、集団（レコード）以上の構造に全て関与している。集団関係（SET）は集団間の関係であり、2組の集団の一方を親集団、他方を従属集団と呼ぶ。網構造とはn組の親集団に、一つの従属集団を許すような構造である。ファイル間、エントリ間に集団関係を定義することができる。

表－1　構造の型

		下部構造
a	項目	なし
b	集団	a, b
c	集団関係	b
d	エントリ	b, c
e	ファイル	d, c
f	データベース	e, c

一つの例として、病院の入院状況に関するDBを示す。英文字(a~j)のついた矢印が集団関係を表わす。両方向に矢印のあるのは、実は別な関係のあり得るのを省略して一本の線にしたためである。本構造でこのようなデータ構造を表現するのは難しく、幾本もの木に分割し、データの重複をしなければならない。Cはある病棟に入っている患者を関係づけているので、現在の入院患者リスト等が分る。またeは二つにわけe1;ID→病名, e2;病名→IDとすれば、e1はある患者の持つ病名(1つとは限らない)が分り、e2はある病名にかかっている患者を全て知ることができる。言うまでもなくこれは冗長であり、e2一つあれば充分である。この場合、ID→病名を探すのは、IDに含まれる病名1, 病2, 病名3のポインターの終点、(あるいは始点)を探すのと同等だからである。

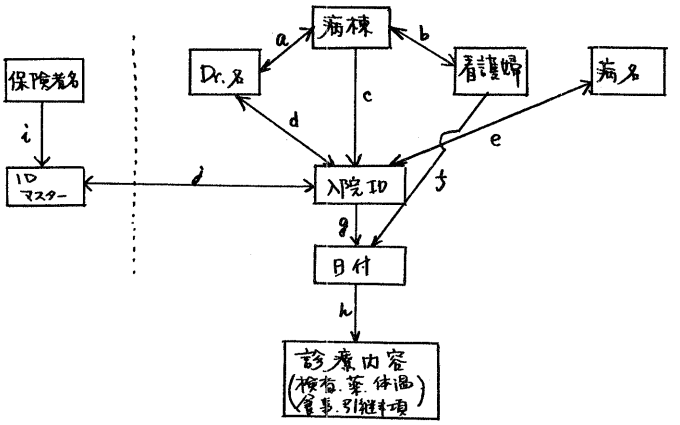


図-2 入院状況DBの例

さて、このようなSETを定義するにあたっては、記憶モードを指定することができる。DBTGではCHAINとPOINTER ARRAYのどちらか一つを選択できる。今、病名とIDの間の関係e2をDBTGに従って定義すると、

```
SET NAME is e2
MODE IS CHAIN ; ORDER IS LAST
OWNER IS 病名
MENBER IS 入院ID OPTIONAL AUTOMATIC
```

となる。ORDER句は新しいレコードを集団のアセンブリのどこに挿入するか決める方法を示す。FIRST, LAST, NEXT, PRIORの4種があり、先頭集団より前、最後の集団の後、現在の集団の直後、直前を意味する。なおSORTEDを指定することもできる。(ex. ORDER IS SORTED WITHIN REDODE NAME.)。図-3にCHAINの例を示す。入院IDにはやや無理がある。同一患者を見るDrが二人いる場合ポインターの位置を識別する'を用いている。CHAINでなくPOINTER ARRAYにした場合の例を図-4に示す。この場合、後属集団がポインターを持つ必要はない。図-4はDr名だけのポインタアレイ

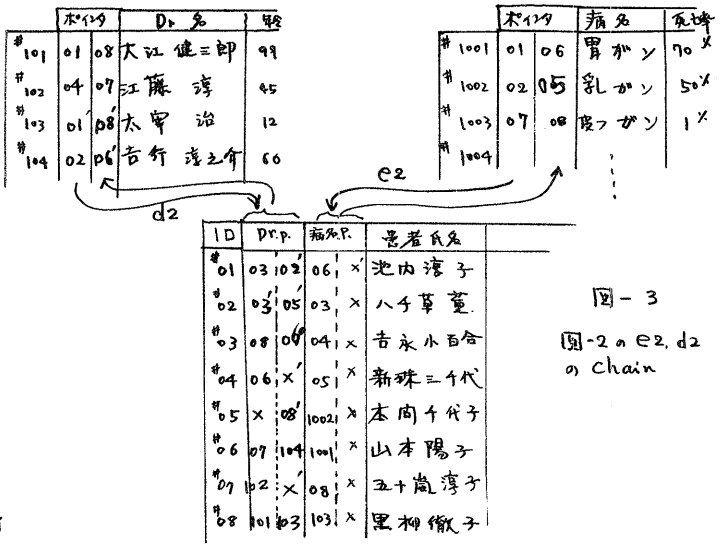


図-3

図-2のe2, d2のchain

である。

DBTG提案のDDLのうち集団関係スキーマを述べたが、スキーマDDLの特長は以下の如である。

#101	大江健三郎	01	03	08		
#102	江藤淳	04	06	07		
#103	太宰治	01	02	05	08	
#104	古行淳之介	02	03	09		

図-4. D名のポインタ・アレイの例

① PRIVACY LOCKが、どの構造の型にでも定義でき、また操作機能（STORE, GET, MODIFY）に対するLOCKがかけられる。
② 上述の如く、集団関係を陽に定義している。
③ レベルによって、ある程度集団スキーマのデータ構造を定義する。

一方、サブ・スキーマ用のDDLとしてはCOBOLを親言語としている。一般にサブ・スキーマは、

① データベース中の主定義から、プログラムが関与するエントリまたはその一部を選ぶ。
② データベースの構造、様式、表現形式、その他の一般的なデータ形式を、親言語のデータ記述方式にあったやり方で記述する。
③ データ項目の表現やレコードの内部構造を記述する主記述（スキーマ）とサブスキーマとの間で対応づけを行う。

ことを行う。サブ・スキーマとスキーマでは次のような差異が生じてよい。

① 項スキーマの性質は項目サブ・スキーマの性質と違って良い。これらの型の対応づけはDMLのGET, STORE, MODIFY命令が行う機能の一つである。
② 機密保護は構造の場所によって異なって良い
③ 要素の順序入れ換え可
④ 項目スキーマ、集団スキーマからなる集団スキーマを構成してもよい。また一つの集団スキーマを分割してもよい。
⑤ 一次元構造の繰り返し集団スキーマを階層構造にしてもよい。（高々3段）
⑥ 集合選基準を変更できる
⑦ 新たな領域やエントリーの新たな呼びあし制限が定義できる。

サブ・スキーマ特有の機能がある。

LOCKS　項目集団などの機密保護ロックの呼びあしに対する制限
DISPLAY　サブスキーマの表示に対する制限
COMPILE　サブスキーマを使うプログラムの制限
ALTER　〃　の変更に対する制限
COPY　スキーマの一部をコピーする。

4.7 DML

以上のDBにアクセスするための言語として、

制御用；OPEN（ファイルを開く）, CLOSE（ファイルを閉じる）, IF（判断）
検索用；FIND（位置決め）, GET（呼び出し）, KEEP, FREE（保持）, MOVE（現状の復帰）
修正用；STORE（追加）, MODIFY（変更）, DELETE（削除）, ORDER（順序づけ、分類）, INSERT, REMOVE（再編成）

の三種に大別できるものがある。先のSETの定義の際、PRIOR, NEXTが定義できることになっているが、それは、実行単位の継続中、システムが実行単位が

最も最近処理した集団を識別する何んらかの情報を更新保持していることが、前提として考えられている。しかし、これは通常、DDLにもDMLにも関係なくOSがサポートするものである。DMLはそれ自体明解であるが、使用するOSに強く依存しているので、OSとして暗黙のうちに仮定している内容を列挙する。

1) 現状（CURRECY）

実行単位の継続中システムは最も新しい処理対象を識別する現在情報を更新保持する。この現在情報とは次に述べるものの Data base key そのものである。①各集団名の Current Record，②各集団関係の Current Record，③扱っている領域内のCR，④実行単位が扱っているRC．⑤ ④が属する集団スキーマ名，⑥ ④が入っている Area Name.

2) UWA（User's working Area）

ここはプログラム毎に持つ領域である。DBの内容をサブ・スキーマの記述で規定されたUWAへ移せるのはGET命令だけである。

3) ERROR，や例外時の内容の保持

誤った状態について，次の六項目が使われる。① ERROR-STATUS 最初に検出された誤りの型とそれが生じたDML命令を示す項目，② ERROR-SET 誤りが発生した集団関係，③ ERROR-RECORD．誤りの発生した集団名，④ ERROR AREA；誤りが発生した領域名，⑤ ERROR TYPE 同時並行実行単位間の干渉による誤り．⑥ ERROR-COUNT；一つの DML命令を実行する間で検出された誤り総数，を保持する．

4) 機密保護処理と保全

DDLで定義された機密保護の他に，二つ以上のプログラムが同時平行処理されている場合に，一方のプログラムが他方の処理に関係なくデータの内容を変えてしまったり，削除してしまうことに関してルールを設定することができる。即ち．①領域を専有して検索する，②他から更新されないよう保護（protected）して検索したり．③専有して更新したり．④保護して更新する，ランク付をOPEN命令時に行うものである。

さてDMLを用いて次のような処理を行ってみよう。（図－3参照）

(1) 吉永小百合の病名．主治医を知る場合

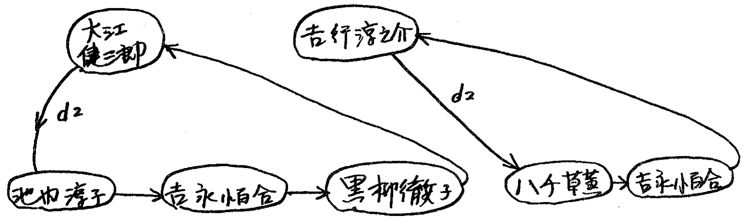
OPEN ALL FOR SET e2，d2

USAGE MODE IS PROTECTED RETRIEVAL

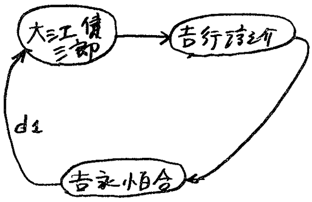
として．保護した検索をするようにSet e2，d2の全てをUWA用に開く。次いで，FIND命令によってCurrent Recordの指示子を更新する（FIND USING #03（#03をIDとする））。更らに，

FIND OWNER in e2 OF CURRENT OF e2 SET
　　　同上　　d2　　　同上　　　d2 同上

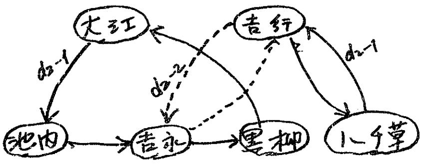
とすることによって．彼女の病名が乳ガンで，主治医が大江健三郎と江藤淳であることが分る。但し，d2を使って2人のDr名を同時に探すことはできない。これは極めて重要な制限で，先の例でもDDLでd2についての定義を示さなかったのはこのためである。主治医が二人いるというような関係はネットワーク構造でも通常許されていない。別なSETを定義するか．d1；ID → Dr名のような入院IDを親とするSetを定義しなければならない。あるいは入院IDを重複して所有しなければならない。実現可能な構成を図－5に示す。ともかく，FIND命令で分るのはDB内，システムバッファ内であり，次のGET命令によって，初



(a) 入院IDを重複する型
Setはd2,1つでより。



(b) Dr名を重複する型
d1だけ



(c) d2-1, d2-2, なるSetを二つに分けて持つ場合

(a) 定義は簡単であるが検索が大変
(b) 〃 検索（主治医の持つ患者を探すような検索）は大変
(c) 定義を2つ行い、2つの定義について検索が必要

図-5. 親が2つ、子がn個の実現例

めてUWAにデータが転送される。

(2) 新たにID #09の十朱幸代が入院してきた場合

STORE命令によって挿入されるが、この命令実行の前に、関連するSet、集団（RECORD）を初期化していなければならない。またDDLでPRIVACY LOCKがかかっている場合には、それらもパスする情報をUWAに入れなければならない。先に示したe2のORDER句で示したように、彼女が皮フガンの場合、図-3の一部が図-6のようになる。

	入院ID				
#08	101	09	09	×	黒柳徹子
#09	103	×	103	×	十朱幸代

	病名			
#1003	07	09	皮フガン	1%

図-6 追加後のレコード

4.8 医療との関連

病院内の医療情報は重複、転記が多く、また種々の使われ方をするという意味で、DBでそれらを実現することが可能である。しかし、DBをよりよく生かすには、①大容量高速2次記憶装置，②多数の端末、③DBとしての利用頻度が多いことが、実現の条件と考えられる。システムの有効性という観点から見ると、その利用価値の認められるところは、割合限られた範囲になるかもしれない。又図-2の例に示したように、ネットワーク形が適していることは正しいと思われるが、どのようなシステム内部構成であっても、（いかにデータ独立性がなく、プログラムオリエンテッドなファイルが管理されていても，いかにプログラマが大変でも、いかにシステム拡張が時間のかかるものであっても）外面的な様相は余り変ったことにはならないので、医療システムの鍵を握るDr、看護婦、管理者事務職員に納得の得られる時間は、相当長期に渡るものと考えられる。